



Universidade Federal do Ceará - Campus de Quixadá
Disciplina: Sistemas Distribuídos
Código: QXD0043
Professor: Rafael Braga

Trabalho 1 – Streams e Sockets

- O trabalho deve ser feito individualmente. Cada estudante deve resolver todas as questões e o código fonte do trabalho deve ser enviado para correção pelo Moodle através da Atividade com título: Trabalho 1 – Streams e Sockets.

- Modele as cinco situações descritas abaixo em classes java utilizando as classes também as classes apresentadas em sala para transmissão de streams em UDP ou TCP.

1. Um registro de uma pessoa é constituído de um nome, seu CPF e sua idade. Crie uma classe Pessoa que represente esta informação. Crie uma subclasse de OutputStream chamada PessoasOutputStream que envia os dados de um conjunto (array) de Pessoas, seguindo a seguinte regra:

- a) Primeiro deve ser enviado o número de pessoas que terão dados enviados pelo stream;
- b) Depois para cada pessoa, deve ser enviada o número de bytes utilizados para gravar o nome da pessoa, o nome da pessoa, seu CPF e sua idade. O nome é uma string e os outros são inteiros;
- c) O construtor da subclasse deve receber como parâmetros: (i) um array de Pessoas que representam os dados a serem transmitidos e um OutputStream de destino, para onde as informações do conjunto de pessoas devem ser enviadas.
- d) Teste a sua implementação utilizando como destino a saída padrão da console (System.out).
- e) Teste a sua implementação utilizando como destino um arquivo (FileOutputStream).

2. Crie uma subclasse para InputStream chamada PessoasInputStream que lê os dados gerados pelo stream do exercício anterior.

- a) O construtor da subclasse deve receber como parâmetro um InputStream de origem, de onde as sequências de bytes serão lidas.
- b) Teste a sua implementação utilizando como origem a entrada padrão da console (System.in).
- c) Teste a sua implementação utilizando como origem um arquivo (FileInputStream).

3. Dado o projeto fornecido de um Banco, implemente os métodos:

```
public void gravarContasArquivo(String nomeArquivo) throws IOException e  
public void carregarContasArquivo(String nomeArquivo) throws IOException
```

da classe BancoContas. O primeiro deve salvar em um arquivo o conteúdo das contas armazenadas em memória. Já o segundo deve carregar os dados salvos no arquivo para memória, simulando a ideia de um banco de dados.

4. Crie uma subclasse para Writer chamada InvertCaseWriter que transforma todos os caracteres escritos no stream da seguinte forma:

- a) Caracteres originalmente enviados em caixa-alta devem ser modificados para caixa-baixa. Caracteres originalmente enviados em caixa-baixa devem ser modificados para caixa-alta;
- b) Todos os outros tipos de caracteres (sinais de pontuação, símbolos, etc) devem ser mantidos inalterados.
- c) O construtor da subclasse deve receber como parâmetro um OutputStreamWriter de destino, para onde os caracteres modificados serão escritos.
- d) Teste a sua implementação utilizando como destino a saída padrão da console (System.out).
- e) Teste novamente sua implementação utilizando como destino um arquivo (FileOutputStream)

5. Crie uma subclasse para Reader chamada InvertCaseReader que leia os dados enviados pelo exercício anterior.

- a) O construtor da subclasse deve receber como parâmetro um OutputStreamReader de origem, para onde os caracteres invertidos serão lidos.
- b) Teste a sua implementação utilizando como destino a entrada padrão do teclado (System.in).
- c) Teste novamente sua implementação utilizando como origem um arquivo (FileInputStream)

Bom trabalho e ótimas festas!